Fringe Projection Profilometry for 3DAI Stage One

Middle Tennessee State University

Department of Engineering Technology

Benjamin Wallace

January 2, 2026

Submitted to Dr. Zhang for research purposes
Department of Engineering Technology
Middle Tennessee State University

Abstract

This report presents the preliminary findings of FPP (Fringe Projection Profilometry) for 3DAI project stage one. The objectives were to use existing hardware to setup a system to capture FPP, convert FPP data into point cloud data with accurate dimensions, and automate the entire process using python. The methodology involved using a Kinect model 1520 to capture images, a Texas Instrument DLP projector to project fringes, and a Windows 10 desktop PC to process and display the data. Preliminary results indicate that automating the process with Python scripts is relatively straight forward, and that accuracy, processing time, and control of ambient conditions may need to be carefully balanced to be useful for AI decision making for a robot cell. The report concludes with recommendations for improving the image quality to improve accuracy of point cloud data without adding additional cameras or fringe projections as this would increase processing time.

Table of Contents

Abstract **1**

1 Introduction **3**

1.1 Objectives 3

2 Methodology **3**

3 Results **5**

4 Discussion **5**

5 Conclusion **6**

1 Introduction

This experiment is setup to explore the potential of using a FPP system to scan items on a conveyor belt and feed the results into an AI model to make sorting decisions. This requires an automated image capturing and processing setup.

1.1 Objectives

- To use existing hardware to setup a system to capture FPP.
- Convert FPP data into point cloud data with accurate dimensions.
- Automate the entire process using python.

2 Methodology

- Materials and equipment.
  - Kinect model 1520
  - TI Projector
  - Windows 10 PC
- Procedures followed.
  - Find suitable python repository for handling fringe projection profilometry calculations.
  - Setup camera and projector.
  - Setup python environment.
  - Write script to calibrate data.
  - Write scripts to collect and process data.
  - Run calibration.
  - Use calibration to capture point cloud data from an object.

To find suitable repository Github was explored for possible open source options. The first promising repository was Structured Light Python by Anton Poroykov and Nikita Sivov. This at first appeared to be flexible enough to use with the setup, but ultimataly proved to be too hard coded to their specific setup.

To address this a second repository was researched. FPP Tools by Nathan Hagen. This repository contains the functions needed to generate the fringe projections and calculate phases. It also includes examples and is well documented. It does not have a general method included for converting height in radians to physical units. It suggests using the formula $z = \frac{L_0 \Delta\phi}{\Delta\phi - 2\pi f_0 d}$ which requires accurate measurements of the camera, projector, and reference plane. Instead a quadratic approximation $z = a(\Delta\phi)^2 + b(\Delta\phi) + c$ where a, b, c are found empirically is used. A single set of coefficients were found for the entire image. This method is compared favorably to other methods by Shijie Feng et al (2021) in Calibration of fringe projection profilometry: A comparative review. The only difference being using a per pixel calibration map as opposed to a global calibration.

Given that calibration would be done using empirical measurements, the exact placements dimensions of the camera and projector were not important. This would be useful for incorporating into a robot cell where calibration could be conducted by the robot holding a flat reference plate and moving a know distance closer to the scanning system. The robot cell setup would be imaging from the top, so a bench setup with the camera and projector above the object to be scanned was setup (See fig. 1)

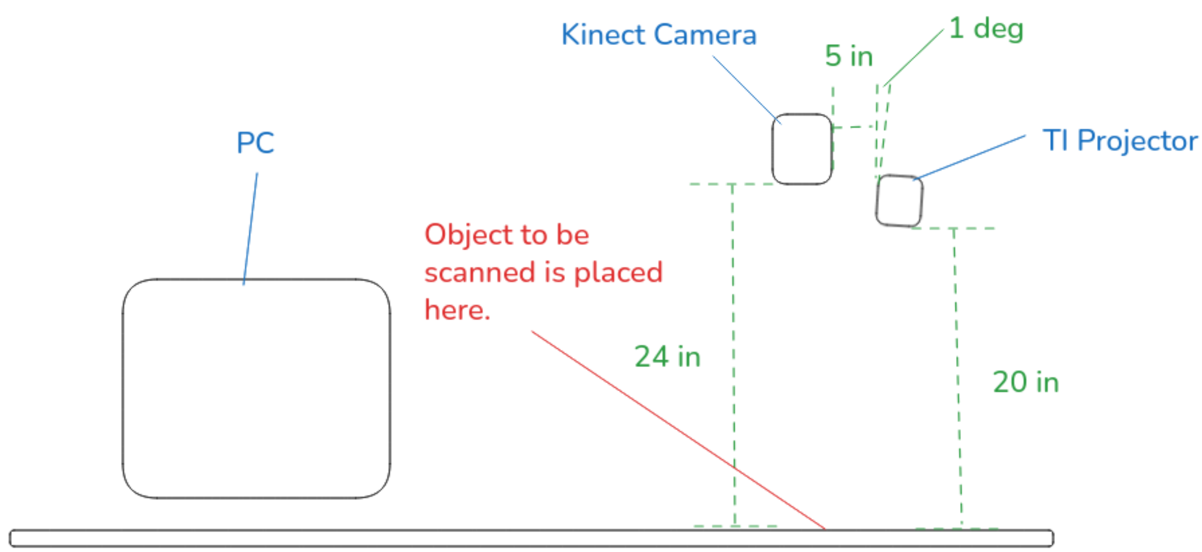

Figure 1: Camera and Projector Setup Diagram

A calibration script was written in Python to simplify the process of finding the coefficients. This script automatically displays the fringes, captures images, and process the data. The user runs the script and predefined height flat plates are requested to be placed into the scanning area. For this experiment, a series of five 2mm thick acrylic sheets where used as the known thicknesses. Each predefined height was thus an additional 2mm, starting at 0mm (reference plane), and ending at 10mm. After the data was collected for the coefficients a, b, c were found using the 'polyfit' function from the 'numpy' library.

Finally a script was written to capture an the FPP of the reference plane and the object. It then calculates the phase shift per pixel and converts that to a height in millimeters using the coefficients found during calibration.

3 Results

A scan of a phone was performed (see fig. 2). The actual height of the phone is about 11.1mm.

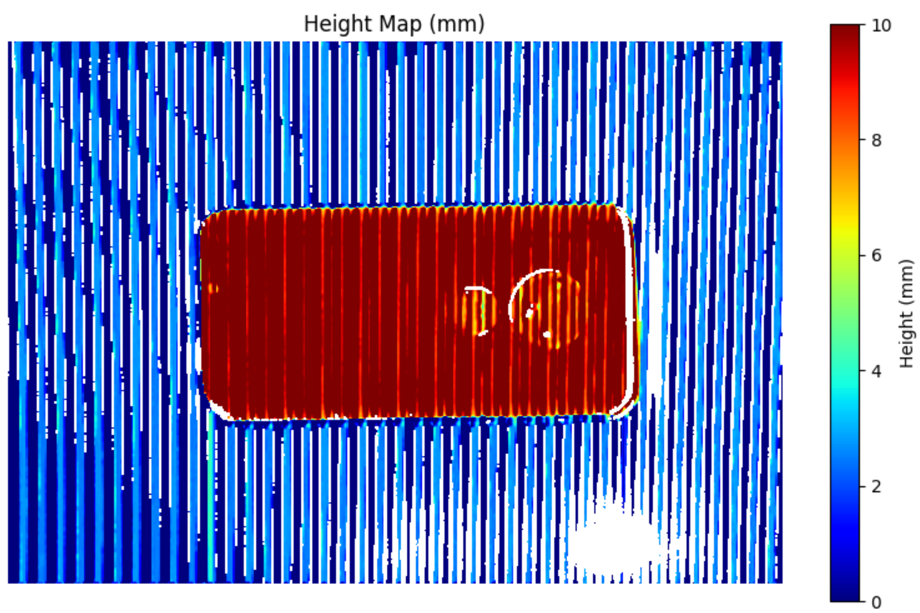

Figure 2: Phone Scan

4 Discussion

The preliminary results of this process look promising. It confirmed the viability of automating the process in a way that could be easily incorporated into a robot cell. There is a wave distortion present in the final height map, but the overall dimension are accurate. The wavy distortion appears most likely to be caused by inconsistency in the pixel brightness of the images captured due to timing between the projector and rolling shutter of the Kinect camera.

5 Conclusion

The primary objectives of setting up a basic system with existing hardware was completed, along with converting the phase data into physical height measurements. Automating the entire process was mostly completed through Python, the path to be able to integrate the system with a robot cell was made much clearer through the experiment.

To further develop this system the accuracy of the height map needs improvement. The main issue being the wavy distortion in the data. The best path to remove reduce this problem appears to be setting up the projector to only project a single color, then capture an image, extract only the color that is projected and process that image. Further enhancement could be achieved by doing per pixel calibration, using better calibration plates (the acrylic sheets are not highly flat, especially when stacked), adding addition fringe projections (currently only using 4), adding a gray scale calibration series to the capture sequence. The enhancements will have to be balanced in this project with the time it takes to process the data to get the level of detail needed for accurate decisions by the AI model.